ACTIVE TESTING OF LARGE LANGUAGE MODELS VIA APPROXIMATE NEYMAN ALLOCATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) are showing superb performance from general-QA to agentic tasks, mathematical frontier and even self-improvement in niche areas. Evaluating these models on domain-specific tasks often requires expert annotation, making it costly to label the entire evaluation pool. Active testing aims to alleviate this bottleneck by approximating the evaluation result from a small but informative subset of the evaluation pool. Prior active-testing approaches are either built for traditional deep learning models or restrict LLM answers to a single token. We remove this restriction and introduce a training-free active testing algorithm for modern reasoning LLMs on tasks with objectively verifiable answers. Our method leverages semantic entropy from surrogate models to stratify the evaluation pool and then conducts approximate Neyman allocation based on signals extracted from these surrogates. Across multiple language and multimodal scenarios, several surrogate-target pairs and different label budgets, our method attains an average 87.0% relative MSE against Uniform Sampling, translating to 15.0%–28.0% label savings.

1 INTRODUCTION

Four layers of large language model (LLM) scaling—pre-training, post-training, test-time, and agentic paradigms (Hoffmann et al., 2022; Ouyang et al., 2022; Yu et al., 2026; Snell et al., 2024; Zuo et al., 2026; Yao et al., 2022; Zhu et al., 2025) — have substantially expanded frontier models' capabilities, from knowledge-based question answering to complex reasoning, multimodal understanding, agentic tasks, and exploring frontiers of science. Reliable evaluation is needed repeatedly throughout the LLM lifecycle, including model selection during pre- and post-training, regression testing across successive releases, and safety and quality monitoring after deployment. While some general capabilities of LLMs are always assessed on publicly available labeled benchmarks, additional evaluation on domain-specific tasks is also essential to better align with downstream applications, where established benchmarks are often absent. Sometimes, obtaining such ground-truth labels costs approximately USD 95 per hour per labor (Rein et al., 2023).

Active testing has been proposed as a principled approach to this regime, selecting a small but informative subset for labeling and using an estimator to recover full-pool performance (Kossen et al., 2021; Farquhar et al., 2021; Berrada et al., 2025). Early approaches are built for traditional deep learning models (Fogliato et al., 2024) and require retraining a data selector (Kossen et al., 2021; 2022), while the LLM extension of Berrada et al. (2025) restricts answers to a single token. For modern reasoning LLMs, this restriction prevents intermediate reasoning before the final answer, changing the model behavior being evaluated. We first try to naively remove this restriction, and study its underlying failure modes in Section 4.1.

In this paper, we propose a training-free and theoretically unbiased active-testing method applicable to modern reasoning LLMs on tasks with verifiable answers. Our motivation is that the compute cost of repeated inference on a small surrogate is a favorable tradeoff to yield an informative signal for evaluating the larger target model under a certain labeling budget. Our method leverages *semantic entropy* (Kuhn et al., 2023; Farquhar et al., 2024) that reflects generative ambiguity in model outputs. Building on this signal, our method constructs a stratified estimator equipped with a surrogate-driven Neyman-style allocation (Neyman, 1992), while remaining fully training-free and compatible with API-based models. The selected operating points in Fig. 1 illustrate the gains.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

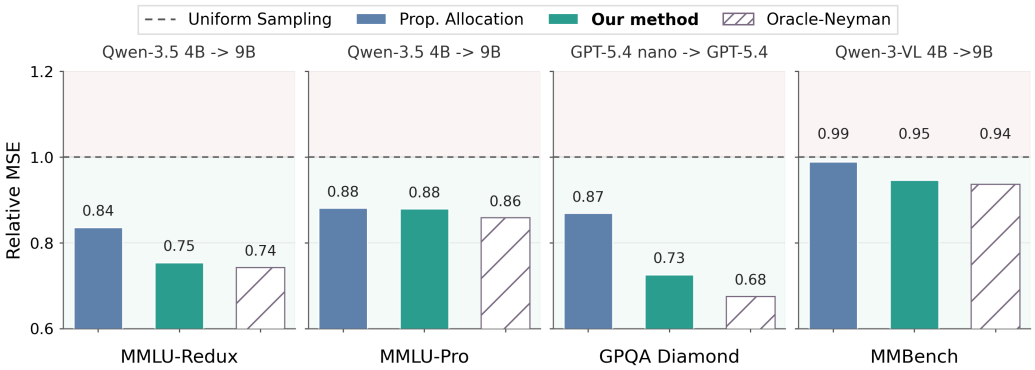

Figure 1: Performance comparison across four benchmarks with labeling budget M=70 (*relative MSE; lower is better*). Our method outperforms the Uniform Sampling baseline and closely approaches the Oracle-Neyman (theoretical optimum), yielding consistent gains across different modalities, multiple benchmarks, and both representative Qwen and frontier GPT models.

Our contributions. Our main contributions are as follows:

(i) To the best of our knowledge, our method is the first active-testing pipeline that provides a label-efficient estimator for reasoning LLMs on tasks with objectively verifiable answers, while simultaneously remaining unbiased, and supporting training-free operation and API-only deployment.

(ii) We study the failure modes of prior arts' extensions, and develop a new algorithm that leverages surrogate-driven semantic entropy to structure the sampling process, couples this signal with a Neyman-style allocation strategy, and yields a novel approach for evaluating both language and multimodal models, with comprehensive ablations to highlight our design choices.

(iii) We conduct extensive experiments across multiple language and multimodal scenarios with both open-source and advanced proprietary models. Among extensions of prior art, our method delivers the best average relative MSE (87.0%) against Uniform Sampling, translating to 15.0%–28.0% savings in expert labels.

Paper organization. The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 introduces the problem formulation. Section 4 presents our method with its motivations. Section 5 reports the main experimental results and additional analyses. Section 6 concludes the paper. Implementation details, additional ablations, and additional experimental results are deferred to the Appendix.

2 Related Work

Active testing. Active testing was introduced by Kossen et al. (2021) for label-efficient model evaluation, training a data selector debiased by the LURE estimator (Kossen et al., 2022; Farquhar et al., 2021). It was later extended to LLMs by Berrada et al. (2025) via a small and frozen LLM surrogate that removes the retraining loop. However, this LLM extension restricts the output to a single token, which is fundamentally incompatible with modern reasoning LLMs that typically generate long chains of thought to exhibit their capabilities.

Fogliato et al. (2024) study a similar surrogate-driven strategy for Computer Vision classifications. AcTracer (Huang et al., 2026) performs multi-stage sampling which relies on hidden states unavailable in black-box APIs, and its confidence-matching proxy does not yield a theoretically unbiased estimator. GAT (Ramakrishnan et al., 2026) retains the LURE estimator but rewrites each input into a True/False proxy task with first-token acquisition, again mismatched with multi-token generative evaluation. Zhou et al. (2025) study efficient pairwise win-rate estimation between two LLMs via control variates,

whereas we target single-model evaluation. Other works (Mozer et al., 2026; Polo et al., 2024; Tahan et al., 2024) also explore label efficiency in evaluation along orthogonal directions.

Sampling, stratification, and budget allocation. Sampling theory provides classical tools for estimating population quantities from a small queried subset (Cochran, 1977; Lohr, 2021). Variance-reducing designs include stratified sampling (Dalenius & Hodges Jr, 1959; Sethi, 1963; Allee & Hidirogloui, 1988), unequal-probability sampling (Hansen & Hurwitz, 1943), systematic sampling (Madow & Madow, 1944), cluster and multi-stage sampling, and adaptive designs (Thompson, 1990). The companion question of budget allocation across strata determines how labels are distributed, ranging from equal allocation and proportional allocation to variance-optimal Neyman allocation under known stratum standard deviations (Neyman, 1992). Such designs admit unbiased finite-population estimation through inverse-probability weighting (Horvitz & Thompson, 1952).

Active learning. Efficient utilization of labeling resources is also a central objective in model training, alongside broader efficiency considerations (Wang et al., 2024a; 2025; Li et al., 2025). Active learning aims to select the most informative subset from an unlabeled pool for annotation, thereby reducing labeling costs (Settles, 2009). Traditional active learning methods are typically based on either diversity (Sener & Savarese, 2017; Agarwal et al., 2020) or uncertainty (Tong & Koller, 2001; Scheffer et al., 2001). It has been shown to improve data efficiency from both theoretical perspectives (Krishnamurthy et al., 2019; Puchkin & Zhivotovskiy, 2021; Zhu & Nowak, 2022a;b) and empirical studies (Saran et al., 2023; Zhang et al., 2023). Recently, active learning has also emerged as a common strategy for improving data efficiency in large pretrained models (Bhatt et al., 2024; Yuan et al., 2024; Zhang & Zhu, 2025).

3 Problem Setting

Active testing. We study active testing for large language or multimodal models on tasks with objectively verifiable answers. The model may generate a multi-token reasoning response with its final answer; the ground truth is labeled and revealed only for queried inputs during active testing. We denote f as the target model including its generation and answer-aggregation protocol (Appendix C), and let $\mathcal{D} = \{x_i\}_{i=1}^{N}$ be a finite pool of unlabeled test inputs drawn from an evaluation distribution p_{eval}.

We denote ℓ as the loss function of predictive errors. Following Berrada et al. (2025), the ideal population risk of interest is:

$$R \;=\; \mathbb{E}_{(x,y)\sim p_{\text{eval}}}[\ell(f(x), y)].$$

Since p_{eval} and the labels of all possible test inputs are unknown, we condition on the fixed pool $\mathcal{D}$ and instead estimate the finite-pool risk:

$$R_{\text{D}} \;=\; \frac{1}{N}\sum_{i=1}^{N} \ell\big(f(x_i), y_i\big).$$

Active testing avoids labeling the full pool by selecting a subset $\mathcal{S} \subset \mathcal{D}$ of size $M \ll N$ and constructing an estimator $\widehat{R}$ for R_{D} from the labeled subset (Kossen et al., 2021). The objective is to minimize the estimation error of $\widehat{R}$ under a fixed labeling budget while preserving unbiasedness with respect to the full-pool risk, $\mathbb{E}[\widehat{R}] = R_{\text{D}}$.

Surrogate and labeling protocol. A small but helpful surrogate model g may generate responses for every input in $\mathcal{D}$ before sampling. These responses may provide auxiliary signals for the active testing even without labeling. Only inputs selected then by active testing algorithm require ground-truth annotation for the risk estimate.

Evaluation metrics. Following Berrada et al. (2025), we evaluate an active-testing method by the squared error of its risk estimate against the northstar finite-pool risk R_{D}, where R_{D} is obtained by labeling the entire pool. For a fixed labeling budget M, let $\widehat{R}^{(t)}$ denote the estimate produced by

estimator $\widehat{R}$ in Monte Carlo trial t, and let $T = 3{,}000$ be the number of trials. We compute the mean squared error (MSE)

$$\mathrm{MSE}(\widehat{R}) = \frac{1}{T}\sum_{t=1}^{T}\left(\widehat{R}^{(t)} - R_{\mathrm{D}}\right)^2.$$

To enable comparisons across different methods, we also follow Berrada et al. (2025) and report the relative-MSE to the Uniform Sampling, which serves as a plain and naive way of active testing. Let $\widehat{R}_{\mathrm{unif}}$ denote the risk induced by Uniform Sampling. We report the relative-MSE ratio $r(\widehat{R}) = \frac{\mathrm{MSE}(\widehat{R})}{\mathrm{MSE}(\widehat{R}_{\mathrm{unif}})}$.

4 METHOD

In this section, we first present the motivation of our approach in Section 4.1. Building on this, we introduce the proposed method and discuss its standard unbiasedness property in Section 4.2.

4.1 MOTIVATION

Berrada et al. (2025) provide an insightful solution of active testing for LLMs though under the restriction of a single-token answer. A natural way to extend that to multi-token reasoning settings, is to retain their LURE estimator (Farquhar et al., 2021) and replace first-token predictive entropy (PE) with token-averaged PE computed over multi-token generations under the same surrogate-driven framework. However, this straightforward PE-based adaptation does not outperform Uniform Sampling, as shown in Table 1.

Table 1: Signal and estimator comparison at $M = 100$, comparing predictive entropy (PE) and semantic entropy (SE) with LURE and stratified Horvitz–Thompson (HT) estimation. Relative MSE (%; Uniform Sampling = 100%). Text pairs use Qwen-3.5 2B → 9B; MMBench uses Qwen-3-VL 2B → 8B.

Benchmark	PE-LURE	SE-LURE	PE+HT	SE+HT (ours)
MMLU-Redux	124.3	230.3	99.7	86.5
MMLU-Pro	154.1	209.7	97.5	86.3
MMBench	99.5	344.6	94.0	93.5
GPQA-Diamond	154.0	185.5	101.3	99.4
Four-pair mean	133.0	242.5	98.1	91.4

We identify two factors underlying this failure. *First, token-level PE entangles semantic uncertainty with surface-form variation*: semantically equivalent generations with different phrasings induce different token-level logit sequences, causing PE to be inflated by paraphrastic diversity rather than genuine difficulty (Kuhn et al., 2023; Farquhar et al., 2024). *Second, modern LLMs are systematically overconfident*, collapsing PE into a narrow low-entropy band and weakening the acquisition signal; surrogate and target PE remain only moderately correlated in our analysis, yielding a substantial mass of instances with systematic disagreement, where surrogate and target confidence sharply diverge (Farquhar et al., 2024).

Based on the analysis of the failure factors, we further consider replacing PE with semantic entropy (SE), an answer-level uncertainty signal that we introduce in Section 4.2. However, this SE-LURE estimator also fails across various datasets, as shown in Table 1. This suggests that the failure is not only due to the uncertainty signal, but also to the LURE estimator itself. In particular, LURE applies per-sample importance weights ($v_m \propto 1/a_m(i_m)$) (Berrada et al., 2025), which amplify surrogate–target mismatches. When the surrogate assigns low acquisition probability to instances that the target is actually uncertain about, these weights dominate the estimator. Since such mismatches are systematic rather than rare, improving the entropy signal alone does not reduce variance but instead exposes this instability.

This observation indicates that the core issue lies in the importance-reweighting mechanism, suggesting the need for an alternative sampling and estimation strategy that avoids such variance amplification.

Algorithm 1 Active Testing via Approximate Neyman Allocation

Input: Pool set $\mathcal{D}$, target model f, surrogate model g, label budget M.

Output: Estimate of full-pool performance $\widehat{R}$.

1: **Signal generation.** For each $x_i \in \mathcal{D}$, draw k surrogate generations and let $\widehat{q}_{ic}$ be the empirical frequency of parsed answer class c. Compute semantic entropy $\mathrm{SE}(g(x_i))$ and self-consistency $\mathrm{SC}(g(x_i))$:

$$\mathrm{SE}(g(x_i)) \;=\; -\sum_{c\in\mathcal{C}_i} \widehat{q}_{ic}\log \widehat{q}_{ic}, \qquad \widehat{q}_{ic} = \frac{1}{k}\sum_{j=1}^{k} \mathbb{I}[A(g(x_i)^{(j)}) = c], \tag{1}$$

$$\mathrm{SC}(g(x_i)) \;=\; \max_{c\in\mathcal{C}_i} \widehat{q}_{ic}. \tag{2}$$

2: **Stratification on semantic entropy.** Partition $\mathcal{D}$ into H strata $\mathcal{D}_0, \ldots, \mathcal{D}_{H-1}$ by binning instances according to their semantic entropy $\mathrm{SE}(g(x_i))$, set $N_h = |\mathcal{D}_h|$, and compute the mean self-consistency score in each stratum:

$$p_h \;=\; \frac{1}{N_h}\sum_{x_i\in\mathcal{D}_h} \mathrm{SC}(g(x_i)).$$

3: **Budget allocation.** Allocate m_h labels per stratum:

$$m_h \;\propto\; N_h \cdot \Big(\sqrt{p_h(1-p_h)} + \delta\Big), \qquad \sum_h m_h = M, \qquad 1 \le m_h \le N_h. \tag{3}$$

4: **Estimation.** Sample $\mathcal{S}_h \subset \mathcal{D}_h$ uniformly without replacement with $|\mathcal{S}_h| = m_h$, and estimate full-pool performance:

$$\widehat{R} \;=\; \frac{1}{N}\sum_{h=0}^{H-1} N_h\,\bar{\ell}_h, \qquad \bar{\ell}_h \;=\; \frac{1}{m_h}\sum_{x_i\in\mathcal{S}_h} \ell(f(x_i), y_i). \tag{4}$$

Motivated by this, we turn to a Neyman-style stratified estimator, which allocates the labeling budget in a variance-aware but unbiased manner.

4.2 Our Method

We present our method in Algorithm 1, a training-free active testing framework while maintaining the unbiasedness property.

The proposed estimator constructs SE-based strata from surrogate-derived uncertainty signals, performs a proxy Neyman-style allocation of the labeling budget across strata, and aggregates the per-stratum target losses with a Horvitz–Thompson estimator. We next explain the details of each of the four steps in Algorithm 1.

Signal generation. This step generates uncertainty signals by running a small LLM surrogate model g on the evaluation pool $\mathcal{D}$ to produce k generations per input ($k = 10$ throughout). The generations are used to calculate semantic entropy and self-consistency scores, which are both informative signals for the following sampling and estimation steps, even without any label information. Due to the small size of the surrogate and reusable prefill KV-cache, this step introduces only modest compute overhead.

Semantic entropy addresses the fact that token-level predictive entropy in generative models can conflate uncertainty over *answer* with uncertainty over surface form (Kuhn et al., 2023; Farquhar et al., 2024). For each input x_i, it groups generations into answer-equivalence classes $\mathcal{C}_i$ and computes entropy over the empirical frequencies $\widehat{q}_{ic}$ in Eq. (1), without using any sequence probabilities or logits.

Based on the generated samples, we construct two training-free uncertainty signals: the per-sample semantic entropy $\mathrm{SE}(g(x_i))$ and the self-consistency score (Wang et al., 2022), defined as in Eq. (2). Here $g(x_i)^{(j)}$ is the j-th surrogate generation for input x_i. Per-sample semantic entropy captures the surrogate's uncertainty on sample x_i, while the self-consistency score measures agreement among surrogate generations (higher values indicate stronger consensus and easier instances). It is used to construct the Bernoulli proxy for Neyman allocation described below.

Stratification on semantic entropy. We partition $\mathcal{D}$ into H strata $\mathcal{D}_0, \ldots, \mathcal{D}_{H-1}$ of sizes $N_h = |\mathcal{D}_h|$ based on semantic entropy. In practice, a non-trivial fraction of samples satisfy $\mathrm{SE}(g(x_i)) = 0$, so we assign them to a base stratum $\mathcal{D}_0 = \{x_i \in \mathcal{D} : \mathrm{SE}(g(x_i)) = 0\}$ and apply equal-quantile binning to the remaining samples $\{x_i \in \mathcal{D} : \mathrm{SE}(g(x_i)) > 0\}$. We set $H = 5$ throughout and provide an ablation over H in Section 5.3.

Budget allocation. Given a fixed stratification, classical Neyman allocation $m_h^* \propto N_h \sigma_h$ provides a theoretical optimum that minimizes the estimation variance (Neyman, 1992). However, this optimum depends on the within-stratum standard deviation σ_h of the target loss, which is not directly accessible without labeling the entire pool. To address this, we replace σ_h with a smoothed Bernoulli proxy $\sqrt{p_h(1-p_h)} + \delta$, where $p_h = \frac{1}{N_h} \sum_{i \in \mathcal{D}_h} \mathrm{SC}(g(x_i))$ denotes the mean surrogate self-consistency within stratum $\mathcal{D}_h$. The resulting allocation is defined as in Eq. (3). The offset $\delta > 0$ prevents m_h from collapsing when p_h approaches 1 in the base stratum. We use $\delta = 0.75$ and report an ablation study in Section 5.3.

Estimation. Within each stratum $\mathcal{D}_h$, we draw m_h samples uniformly without replacement and obtain their labels. Let $\mathcal{S}_h \subset \mathcal{D}_h$ denote the selected subset, and define the within-stratum empirical loss mean

$$\bar{\ell}_h = \frac{1}{m_h} \sum_{i \in \mathcal{S}_h} \ell(f(x_i), y_i).$$

Since each sample in stratum $\mathcal{D}_h$ is selected with equal inclusion probability $\pi_h = m_h / N_h$, we can write the Horvitz–Thompson form of $\widehat{R}$ over the globally selected set $\mathcal{S} = \bigcup_{h=0}^{H-1} \mathcal{S}_h$:

$$\widehat{R} = \frac{1}{N} \sum_{i \in \mathcal{S}} \frac{\ell(f(x_i), y_i)}{\pi_i}.$$

Under stratified simple random sampling, $\pi_i = \pi_h$ for all $x_i \in \mathcal{D}_h$, so the Horvitz–Thompson estimator reduces to Eq. (4). This estimator inherits the standard unbiasedness property of stratified estimation (Horvitz & Thompson, 1952; Cochran, 1977).

5 Empirical Results

We conduct comprehensive experiments to examine the effectiveness of our method. We describe the experimental setup in Section 5.1, present the main results in Section 5.2, and provide analyses and ablations in Section 5.3.

5.1 Experimental setup

Datasets. We simulate active testing on existing benchmarks spanning language and multimodal evaluation scenarios. MMLU-Redux (Gema et al., 2025) and MMLU-Pro (Wang et al., 2024b) assess broad academic knowledge and reasoning, GPQA-Diamond (Rein et al., 2023) targets graduate-level science questions, and MMBench (Liu et al., 2024) evaluates vision-language perception and reasoning. Although these benchmarks are open sourced with ground-truth labels, we initially hide them from the active-testing methods. For each labeling budget M, ground-truth labels are revealed only for the M inputs selected during active testing and are used to estimate full-pool performance. The complete labels are used separately to compute the reference full-pool risk for measuring estimation error and to construct the infeasible Oracle-Neyman baseline. Task type, knowledge domain, and pool size for each benchmark are summarized in Table 13 in the Appendix.

Models. We study both large language and multimodal settings using the open-source Qwen-3.5 series (Qwen Team, 2026), Qwen-3-VL series (Bai et al., 2025), and advanced API-based GPT-5.4 series (OpenAI, 2026a;b). These model families cover both controlled open-weight scaling and frontier API deployment settings. For each surrogate-target pair, the weaker or smaller model is used as the surrogate g, while the stronger or larger model serves as the target f. All models generate free-form text, which is subsequently mapped to canonical answers via a parser for evaluation; we do not restrict the target to first-token-only decoding.

Baselines. As discussed in Section 4.1 and shown in Table 1, the PE-LURE and SE-LURE adaptations of prior active-testing methods generally yield higher MSE than Uniform Sampling in the four-pair comparison, so we exclude them from the main results. Since our approach is a surrogate-driven approximation to Neyman allocation, we compare it against three allocation baselines in the main budget curves: *Uniform Sampling*, *Proportional Allocation*, and *Oracle-Neyman*. We evaluate all methods using the MSE and relative MSE defined in Section 3.

Uniform Sampling selects M samples uniformly without replacement for labeling. Proportional Allocation uses the same SE-based stratification as our method, but allocates the labeling budget proportionally to stratum size, i.e., $m_h \propto N_h$. This baseline serves as a within-framework reference: the gap between *Uniform Sampling* and *Proportional Allocation* isolates the effect of semantic-entropy-based stratification, while the additional gap between *Proportional Allocation* and *our method* captures the benefit of our proxy-Neyman allocation in Eq. (3). *Oracle-Neyman* allocates the budget according to the oracle target-side within-stratum standard deviation, i.e., $m_h \propto N_h \sigma_h$. As it relies on inaccessible target statistics, it is infeasible in practice and serves only as a theoretical upper bound on the best achievable allocation under the given stratification.

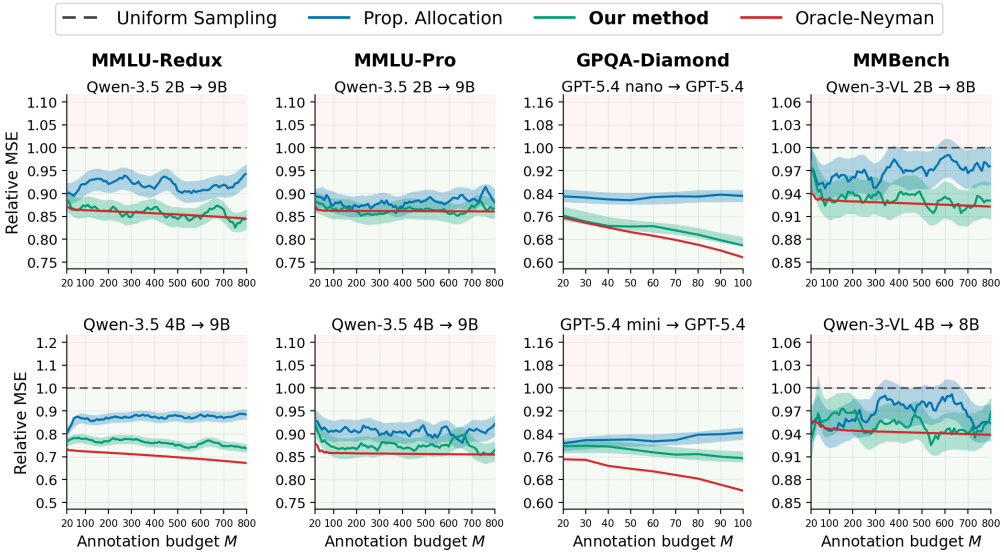

Figure 2: Results across four benchmarks with various surrogate-target model pairs. We report relative MSE (*lower is better*) as annotation budget M increases. Our method consistently achieves better performance while closely matching the Oracle-Neyman (theoretical optimum). Each curve is averaged over 3,000 Monte Carlo trials, with shaded bands indicating ± 1 standard error of the mean.

5.2 Main Results

Our method consistently reduces estimation error. We compare our method with the baselines across budgets, benchmarks and model variants. We present selected scenarios in Fig. 2, and our method remains below *Uniform Sampling* throughout the budget sweep and closely follows the infeasible *Oracle-Neyman* optimum. Beyond that, Table 2 summarizes all evaluated pairs over the common budgets $M \in \{10, 20, \ldots, 190\}$. Our method achieves a superb 87.0% relative MSE against Uniform Sampling across budgets, pairs, and benchmarks, compared with 91.0% for Proportional and 84.7% for the infeasible Oracle-Neyman reference.

Table 2: Relative MSE (%; Uniform Sampling = 100%; lower is better) over all evaluated pairs and common budgets $M = 10, 20, \ldots, 190$. Each benchmark averages over its pairs and budgets; the final row weights benchmarks equally. *Oracle-Neyman* (gray column) is infeasible and is excluded from the per-row best-value comparison. The best value among feasible methods in each row is shown in bold.

Benchmark	Pairs	Proportional	Ours ↓	Oracle-Neyman
MMLU-Redux	3	88.8	**83.4**	81.7
MMLU-Pro	3	88.7	**87.1**	86.2
MMBench	3	95.4	**95.0**	93.7
GPQA-Diamond	8	90.9	**82.3**	77.1
Benchmark-macro mean	17	91.0	**87.0**	84.7

Label savings and compute overhead. At matched MSE, our method translates to 15.0%–28.0% fewer labels in Table 3 (mean: 22.9%). Although expert-annotation and compute costs cannot be universally reduced to one scalar, Rein et al. (2023) estimate PhD-level expert compensation at approximately USD 95/hour, whereas AWS lists a GPU similar to the one we used in practice at USD 1.861/hour (Amazon Web Services, 2026). For the MMLU-Pro Qwen-3.5 2B → 4B pair in Table 3, the surrogate overhead of 7.94 GPU-hours costs about USD 14.78, which is substantially less than the cost of one hour of expert labor, whereas annotating the 74 additional examples required by Uniform Sampling would clearly require substantially more than one expert-hour.

Table 3: Label savings and compute overhead at matched MSE. Qwen inference is priced at $1.861/GPU-hour according to AWS (Amazon Web Services, 2026); for GPQA-Diamond, we estimate additional cost using API pricing.

		Our label saving	Compute overhead	
Benchmark	Surrogate → Target	(labels)	(hours)	($)
MMLU-Redux	Qwen-3.5 4B → 9B	56	3.87	7.20
MMLU-Pro	Qwen-3.5 2B → 4B	74	7.94	14.78
MMBench	Qwen-3-VL 2B → 4B	12	0.154	0.29
GPQA-Diamond	GPT-5.4 Nano → 5.4	8	N/A	2.01

5.3 Analyses and Ablations

We ablate the two central design choices of our method: the stratification method that maps surrogate semantic entropy onto bin boundaries, and the allocation rule that distributes the budget across strata. All ablations are conducted on MMLU-Redux with the Qwen-3.5 2B surrogate against the Qwen-3.5 9B target. Both studies report relative MSE (%; Uniform Sampling = 100%; lower is better).

Selection of stratification method. We compare three traditional stratifications: equal-width, k-means, and quantile binning, with our Adaptive SE stratification, which separates zero-entropy inputs before quantile-binning the remainder. With allocation fixed, Adaptive SE achieves the lowest average relative MSE (84.3%), compared with 86.1%, 84.8%, and 84.7%, respectively (Table 4).

Effectiveness of allocation strategy. We assess the effectiveness of our allocation strategy on MMLU-Redux using the Qwen-3.5 2B surrogate against the Qwen-3.5 9B target. As shown in Table 5, we fix SE-based stratification and compare five allocation strategies: *Equal Allocation* ($m_h = M/H$), *Proportional Allocation* ($m_h \propto N_h$), *Power Allocation* ($m_h \propto \sqrt{N_h}$), *our method* from Eq. (3), and *Oracle-Neyman* ($m_h \propto N_h \sigma_h$) which serves as an infeasible upper bound under the fixed stratification. Two findings emerge here. First, size-only heuristics are weak: Equal (90.9%) and Proportional Allocation (91.4%) retain about 90% of the Uniform Sampling MSE, showing that stratification without variance information yields limited gains. Second, proxy-Neyman performs best at nearly all budgets and closely matches Oracle-Neyman on average (84.3% vs. 82.2%), while Power lies in between (86.3%), indicating that the surrogate captures variance information beyond size alone.

Table 4: Ablation study of Algorithm 1 with different stratification methods (when H=5) on the MMLU-Redux dataset, using Qwen-3.5 9B as the target and Qwen-3.5 2B as the surrogate. We report relative MSE (%, *lower is better*) versus Uniform Sampling and the best performance in each row is highlighted in bold.

Budget M	Equal-width	k-means	Quantile	Adaptive SE (Ours) ↓
50	89.9	87.0	86.1	**85.7**
100	89.3	**82.0**	87.7	83.7
200	84.1	84.8	**81.8**	82.5
400	**84.5**	84.6	84.7	86.9
800	82.7	85.6	83.3	**82.5**
Average	86.1	84.8	84.7	**84.3**

Table 5: Ablation study of our method varying the allocation strategy on MMLU-Redux dataset with Qwen-3.5 2B (surrogate) and Qwen-3.5 9B (target) model pairs. We report the relative MSE (%, *lower is better*) versus Uniform Sampling as the annotation budget increases. *Oracle-Neyman* (gray column) is infeasible in practice and serves as the theoretical optimum given the stratification; it is excluded from the per-row best-value comparison. The best value among feasible rules in each row is shown in bold.

Budget M	Equal	Proportional	Power	Proxy-Neyman (Ours) ↓	Oracle-Neyman
50	95.0	92.6	89.3	**85.7**	84.3
100	95.1	91.9	85.4	**83.7**	82.7
200	86.9	86.6	84.9	**82.5**	82.0
400	90.0	92.3	**86.2**	86.9	81.2
800	87.4	93.4	85.6	**82.5**	80.7
Average	90.9	91.4	86.3	**84.3**	82.2

In Appendix A, we present additional experiments comparing a Fogliato-style study, extending evaluation to verifier-scored responses, and reporting paired effect sizes. Additional ablations cover cross-family surrogates, two-by-two controls, the number of strata, and the allocation offset (Appendix B).

6 Conclusion and Limitations

We studied active testing for reasoning language and multimodal models on tasks with objectively verifiable answers, where each label usually requires costly expert annotation. We introduced a training-free method that uses answer-level uncertainty to partition the evaluation pool, and estimates full-pool risk with a surrogate-driven proxy for Neyman allocation. Across language and vision-language scenarios with open-weight and proprietary models, our method reduces average estimation error relative to Uniform Sampling, closely tracks the infeasible Oracle-Neyman reference, and reaches matched precision with substantially fewer expert labels.

Our study also has some scope limitations. First, our experiments focus on tasks with objectively verifiable answers. Second, poor surrogate–target alignment can reduce or reverse the variance gains relative to Uniform Sampling. Future work can also extend this framework by developing modality-aware surrogate signals and allocation rules for richer vision–language, interactive, and open-ended evaluation tasks.

AI USE STATEMENT

In this work, we used generative AI tools for developing theoretical models or conceptual frameworks, designing or providing feedback on research methodology or experiments, implementing methods, assisting with translation, supporting qualitative and thematic data analysis, and interpreting results. We have not used generative AI tools for formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, or proposing or refining hypotheses, and generating synthetic data sets and cleaning and reformatting datasets are not applicable to this work. Additionally, we used generative AI tools for creating or modifying scientific figures or images, creating or editing software code, summarizing or analysing existing literature, discovering research topics or identifying gaps, brainstorming, editing the paper to improve readability, and formatting references. We have reviewed all AI-assisted work. AI-assisted text and code were checked against the saved experimental records and implementation details. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora. Contextual diversity for active learning. In *European Conference on Computer Vision*, pp. 137–153. Springer, 2020.

PLV Allee and Michel A Hidirogloui. On the stratification of skewed populations. *Survey methodology*, 14(1):33–43, 1988.

Amazon Web Services. Amazon EC2 on-demand pricing, 2026. URL https://aws.amazon.com/ec2/pricing/on-demand/.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Gabrielle Berrada, Jannik Kossen, Freddie Bickford Smith, Muhammed Razzak, Yarin Gal, and Tom Rainforth. Scaling up active testing to large language models. *arXiv preprint arXiv:2508.09093*, 2025.

Gantavya Bhatt, Yifang Chen, Arnav Das, Jifan Zhang, Sang Truong, Stephen Mussmann, Yinglun Zhu, Jeff Bilmes, Simon Du, Kevin Jamieson, et al. An experimental design framework for label-efficient supervised finetuning of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6549–6560, 2024.

William G Cochran. Sampling techniques. *Johan Wiley & Sons Inc*, 1977.

Tore Dalenius and Joseph L Hodges Jr. Minimum variance stratification. *Journal of the American Statistical Association*, 54(285):88–101, 1959.

Sebastian Farquhar, Yarin Gal, and Tom Rainforth. On statistical bias in active learning: How and when to fix it. *arXiv preprint arXiv:2101.11665*, 2021.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Riccardo Fogliato, Pratik Patil, Mathew Monfort, and Pietro Perona. A framework for efficient model evaluation through stratification, sampling, and estimation. In *European Conference on Computer Vision*, pp. 140–158. Springer, 2024.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, et al. Are we done with mmlu? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5069–5096, 2025.

Morris H Hansen and William N Hurwitz. On the theory of sampling from finite populations. *The Annals of Mathematical Statistics*, 14(4):333–362, 1943.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, DDL Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 10, 2022.

Daniel G Horvitz and Donovan J Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260):663–685, 1952.

Yuheng Huang, Jiayang Song, Qiang Hu, Felix Juefei-Xu, and Lei Ma. Actracer: Active testing of large language model via multi-stage sampling. *ACM Transactions on Software Engineering and Methodology*, 35(3):1–30, 2026.

Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Tom Rainforth. Active testing: Sample-efficient model evaluation. In *International Conference on Machine Learning*, pp. 5753–5763. PMLR, 2021.

Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Thomas Rainforth. Active surrogate estimators: An active learning approach to label-efficient model evaluation. *Advances in Neural Information Processing Systems*, 35:24557–24570, 2022.

Akshay Krishnamurthy, Alekh Agarwal, Tzu-Kuo Huang, Hal Daumé III, and John Langford. Active learning for cost-sensitive classification. *Journal of Machine Learning Research*, 20(65):1–50, 2019.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*, 2023.

Zexin Li, Jiancheng Zhang, Yufei Li, Yinglun Zhu, and Cong Liu. Mixtraining: A better trade-off between compute and performance. *arXiv preprint arXiv:2502.19513*, 2025.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024.

Sharon L Lohr. *Sampling: design and analysis*. Chapman and Hall/CRC, 2021.

William G Madow and Lillian H Madow. On the theory of systematic sampling, i. *The Annals of Mathematical Statistics*, 15(1):1–24, 1944.

Reagan Mozer, Nicole E Pashley, and Luke Miratrix. Stratified sampling for model-assisted estimation with surrogate outcomes. *arXiv preprint arXiv:2602.12992*, 2026.

Jerzy Neyman. On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection. In *Breakthroughs in statistics: Methodology and distribution*, pp. 123–150. Springer, 1992.

OpenAI. Introducing GPT-5.4, March 2026a. URL https://openai.com/index/introducing-gpt-5-4/.

OpenAI. Introducing GPT-5.4 mini and nano, March 2026b. URL https://openai.com/index/introducing-gpt-5-4-mini-and-nano/.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinybenchmarks: evaluating llms with fewer examples. *arXiv preprint arXiv:2402.14992*, 2024.

Nikita Puchkin and Nikita Zhivotovskiy. Exponential savings in agnostic active learning through abstention. In *Conference on learning theory*, pp. 3806–3832. PMLR, 2021.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Aashish Anantha Ramakrishnan, Ardavan Saeedi, Hamid Reza Hassanzadeh, Fazlolah Mohaghegh, and Dongwon Lee. Generative active testing: Efficient llm evaluation via proxy task adaptation. *arXiv preprint arXiv:2603.19264*, 2026.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Akanksha Saran, Safoora Yousefi, Akshay Krishnamurthy, John Langford, and Jordan T Ash. Streaming active learning with deep neural networks. In *International Conference on Machine Learning*, pp. 30005–30021. PMLR, 2023.

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Active hidden markov models for information extraction. In *International symposium on intelligent data analysis*, pp. 309–318. Springer, 2001.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

VK Sethi. A note on optimum stratification of populations for estimating the population means. *Australian Journal of Statistics*, 5(1):20–33, 1963.

Burr Settles. Active learning literature survey. 2009.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Shir Ashury Tahan, Ariel Gera, Benjamin Sznajder, Leshem Choshen, Liat Ein Dor, and Eyal Shnarch. Label-efficient model selection for text generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8384–8402, 2024.

Steven K Thompson. Adaptive cluster sampling. *Journal of the american statistical association*, 85 (412):1050–1059, 1990.

Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

Haonan Wang, Zeli Liu, Chao Fang, John Paul Walters, and Stephen P Crago. Moq: mixture-of-format activation quantization for communication-efficient ai inference system. In *NeurIPS 2024 Workshop Machine Learning with new Compute Paradigms*, 2024a.

Haonan Wang, Zeli Liu, Kajimusugura Hoshino, Tuo Zhang, John Paul Walters, and Stephen Crago. Fedpai: Achieving extreme sparsity in federated learning via pruning at initialization. In *International Conference on Intelligent Computing*, pp. 350–364. Springer, 2025.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290, 2024b.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Yue Yu, Bowen Zuo, David J. Crandall, Yinglun Zhu, and Dongruo Zhou. Exploring more, reasoning better: Stepwise risk-sensitive GRPO for diffusion language models. In *Non-Autoregressive Language Models for Fast and Flexible Text Generation*, 2026. URL https://openreview.net/forum?id=7sbmxAatQh.

Bo Yuan, Yulin Chen, Yin Zhang, and Wei Jiang. Hide and seek in noise labels: Noise-robust collaborative active learning with llms-powered assistance. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10977–11011, 2024.

Jiancheng Zhang and Yinglun Zhu. Towards multimodal active learning: Efficient learning with limited paired data. *arXiv preprint arXiv:2510.03247*, 2025.

Jifan Zhang, Yifang Chen, Gregory Canal, Stephen Mussmann, Arnav M Das, Gantavya Bhatt, Yinglun Zhu, Jeffrey Bilmes, Simon Shaolei Du, Kevin Jamieson, et al. Labelbench: A comprehensive framework for benchmarking adaptive label-efficient learning. *arXiv preprint arXiv:2306.09910*, 2023.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Zhaoyi Zhou, Yuda Song, and Andrea Zanette. Accelerating unbiased llm evaluation via synthetic feedback. *arXiv preprint arXiv:2502.10563*, 2025.

Yinglun Zhu and Robert Nowak. Active learning with neural networks: Insights from nonparametric statistics. *Advances in Neural Information Processing Systems*, 35:142–155, 2022a.

Yinglun Zhu and Robert Nowak. Efficient active learning with abstention. *Advances in Neural Information Processing Systems*, 35:35379–35391, 2022b.

Yinglun Zhu, Jiancheng Zhang, and Fuzhi Tang. Test-time matching: Unlocking compositional reasoning in multimodal models. *arXiv preprint arXiv:2510.07632*, 2025.

Bowen Zuo, Dongruo Zhou, and Yinglun Zhu. Adaptive test-time compute allocation with evolving in-context demonstrations. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 35156–35173, 2026.

A Additional experiments

A.1 Fogliato-style study

Although Fogliato et al. (2024) is primarily for Computer Vision models and tasks, we implement its LLM transfer by preserving its k-means design on our surrogate self-consistency (SC), using the same proportional allocation and the same stratified HT. We view this comparison as an inspired design control, not a reproduction of the original. Its benchmark-macro relative MSE is 90.9%, compared with 87.0% for our method. All methods use the same reported benchmark aggregation as Table 2; benchmark-level results appear in Table 6.

Table 6: Adapted Fogliato-style controls for LLM evaluation. Relative MSE (%; Uniform Sampling = 100%; lower is better), averaged over the 19 common budgets and available pairs within each benchmark. The two SC k-means columns use proportional allocation or our SC proxy for Neyman allocation; these are adapted controls, not literal reproductions. Bold marks the best feasible value in each row.

Benchmark	Pairs	SE + Prop.	SC k-means + Prop.	SC k-means + Our proxy	Ours
MMLU-Redux	3	88.8	89.3	87.1	**83.4**
MMLU-Pro	3	88.7	88.6	87.6	**87.1**
MMBench	3	95.4	95.9	96.0	**95.0**
GPQA-Diamond	8	90.9	89.7	88.4	**82.3**
Benchmark-macro mean	17	91.0	90.9	89.8	**87.0**

A.2 Free-form scenario extensions

We conduct experiments on two additional scenarios to show the effectiveness of our method on different settings. IFEval (Zhou et al., 2023) checks instruction-following constraints and groups generations by verifier-outcome tuples. HMMT November 2025 is a Harvard–MIT contest-math pool. Each problem asks for a free-form final answer, but the cluster is defined by an external check while remaining objectively verifiable. Both settings use Qwen-3.5 4B as the surrogate and 9B as the target, with 3,000 Monte Carlo trials per budget. Table 7 records the results.

Table 7: Free-form scenario extensions. Relative MSE (%; Uniform Sampling = 100%; lower is better). *Oracle* (gray column) is infeasible and is excluded from the per-row best-value comparison. The best value among feasible methods in each row is shown in bold.

Benchmark	M	Prop.	Ours ↓	Oracle	PE-LURE	SE-LURE
IFEval	30	75.4	**64.1**	57.6	204.0	189.9
	50	72.8	**67.9**	57.0	203.7	205.6
	100	78.8	**62.7**	51.3	212.1	158.7
	150	84.9	**63.8**	45.7	226.9	127.4
	200	81.0	**60.3**	42.0	218.7	108.2
HMMT Nov. 2025	15	93.7	**85.0**	84.8	100.3	193.8
	20	81.5	**74.2**	72.7	99.7	130.6
	25	63.7	**59.6**	59.0	110.7	85.7

A.3 Paired comparisons across random seeds

The main experiments estimate each method's average MSE over 3,000 Monte Carlo trials. To complement these averages with a trial-level comparison, we evaluate our method and each baseline using the same 3,000 random seeds on each fixed benchmark pool and labeling budget in Table 8. For each seed, we compare their squared estimation errors against the full-pool risk. We report both the relative MSE, which measures the average error reduction, and the win rate, which measures how often our method has smaller squared error than the baseline, while some randomness is introduced by the single seed. Together, these metrics describe the magnitude and seed-level consistency of the gains.

Table 8: Paired comparisons for four surrogate–target pairs using 3,000 shared trial seeds per budget. Relative MSE (%; each indicated baseline = 100%; lower is better). W/R denotes the reported win-rate (↑).

		Proportional		Fogliato-style	
Benchmark	Budget	Rel. MSE	W/R (%)	Rel. MSE	W/R (%)
MMLU-Redux	100	88.6	91.4	91.4	82.3
	Average	91.1	79.3	91.8	76.4
MMLU-Pro	100	99.3	55.4	99.6	54.9
	Average	96.5	68.3	99.2	53.7
MMBench	100	98.0	65.4	96.2	70.1
	Average	97.6	65.5	97.8	65.2
GPQA-Diamond	100	86.3	92.6	82.1	93.9
	Average	83.4	91.1	80.7	88.4

Pairs: Qwen-3.5 4B → 9B for MMLU-Redux and MMLU-Pro; Qwen-3-VL 4B → 8B for MMBench; GPT-5.4 mini → GPT-5.4 for GPQA-Diamond. Fogliato-style uses SC k-means strata, proportional allocation, and HT estimation.

B Additional ablations

B.1 Cross-family surrogates

In Table 9, we report the cross-family ablations on GPQA and GPQA-Diamond, using Qwen as surrogate and GPT as target model. All eight reported pair-budget points improve over Proportional and Uniform Sampling. These results show that a same-family surrogate is not a necessary requirement in our settings, as long as the surrogate is not too weak to provide a useful signal.

Table 9: Cross-family evaluation. Relative MSE (%; Uniform Sampling = 100%; lower is better). The best value among the two methods in each budget is shown in bold.

			$M = 50$		$M = 100$	
Benchmark	Surrogate	Target	Prop.	Ours ↓	Prop.	Ours ↓
GPQA	Qwen-3.5 4B	GPT-5.4	90.0	**88.9**	91.1	**87.6**
GPQA	Qwen-3.5 9B	GPT-5.4	80.5	**75.8**	81.0	**74.8**
GPQA-Diamond	Qwen-3.5 4B	GPT-5.4	91.3	**86.5**	91.7	**81.8**
GPQA-Diamond	Qwen-3.5 9B	GPT-5.4	83.8	**78.0**	84.1	**72.8**

B.2 Two-by-two controls

At $M = 100$, we cross PE and SE with LURE and stratified HT on four surrogate–target pairs in (Table 1). SE+HT lowers the four-pair mean relative MSE from 98.1% for PE+HT to 91.4%, and is lower on every pair. By this ablation, we show that SE does not improve LURE, while stratified HT lowers MSE for both signals. The ablation highlights our design choice that requires SE stratification together with HT estimation.

B.3 Number of strata and allocation offset

We vary the number of strata H and the allocation offset δ on MMLU-Redux with the Qwen-3.5 2B surrogate and Qwen-3.5 9B target, as in Section 5.3. We report relative MSE (%) versus Uniform Sampling (lower is better).

Effect of H. To study the impact of the number of SE-based strata H, we vary H from 2 to 8; results are shown in Table 10. We observe a non-monotonic trend in relative MSE: smaller values of H (e.g., $H = 2$) fail to adequately separate easy and hard instances, leading to high within-stratum variance, while overly large values (e.g., $H = 8$) result in insufficient samples per stratum for reliable

estimation. The best performance is achieved at $H = 5$, which attains the lowest average MSE (84.3% relative to Uniform Sampling). Based on this result, we use $H = 5$ as the default setting in all main experiments.

Table 10: **Ablation on the number of strata H.** Relative MSE (%, lower is better) versus Uniform Sampling on MMLU-Redux, Qwen-3.5 2B surrogate → Qwen-3.5 9B target. Best value per row in bold.

Budget M	$H = 2$	$H = 3$	$H = 5$ (Ours)↓	$H = 8$
50	90.3	87.6	**85.7**	89.3
100	91.3	86.3	**83.7**	83.9
200	86.5	85.5	**82.5**	83.4
400	91.7	**85.4**	86.9	86.9
800	82.6	84.2	**82.5**	83.7
Average (↓)	88.5	85.8	**84.3**	85.5

Impact of offset δ. To evaluate the impact of the offset δ in the proxy-Neyman allocation $m_h \propto N_h\left(\sqrt{p_h(1-p_h)} + \delta\right)$, we sweep $\delta \in \{0.5, 0.75, 1, 2, 5\}$. As shown in Table 11, the sweep reveals a broad optimum: $\delta \in [0.5, 1]$ achieves similar performance, with average relative MSE between 83.9% and 84.3%, while performance degrades to 86.2% at $\delta = 2$ and 87.6% at $\delta = 5$. This trend reflects a trade-off in the surrogate-side proxy: small δ amplifies noise in $\sqrt{p_h(1-p_h)}$ by approaching pure proxy-Neyman allocation, while large δ flattens the allocation toward Proportional Allocation and weakens the variance signal. We adopt $\delta = 0.75$ as the default (84.3%); $\delta = 1$ attains the lowest average relative MSE in the sweep (83.9%).

Table 11: **Ablation on the offset δ in the proxy-Neyman allocation.** Relative MSE (%, lower is better) versus Uniform Sampling on MMLU-Redux, Qwen-3.5 2B surrogate → Qwen-3.5 9B target. Best value per row in bold.

Budget M	$\delta = 0.5$	$\delta = 0.75$ (Ours)↓	$\delta = 1$	$\delta = 2$	$\delta = 5$
50	86.0	**85.7**	85.8	89.0	89.8
100	82.2	83.7	**80.6**	82.3	83.5
200	83.8	**82.5**	82.9	85.3	87.1
400	**86.3**	86.9	87.1	89.7	90.2
800	**81.9**	82.5	83.2	84.6	87.4
Average	84.0	84.3	**83.9**	86.2	87.6

C Reproducibility

Models, data, and code. All datasets are downloaded from Hugging Face using the public benchmark releases in Table 13. For open-weight experiments, we download the official Qwen family from Hugging Face. For API-based experiments, we use the official OpenAI API service. We provide full algorithmic details, hyperparameters, and generation/parsing protocols in this appendix to support reproduction without released code.

Generation and parsing details. Each GPU is used independently for inference with the vLLM framework on NVIDIA RTX 6000 Ada hardware. Surrogate signals are generated with the vLLM decoding configuration listed in Table 12. We parse model answers with the official standard parser or evaluator released by each benchmark. This benchmark-official protocol may differ from model-official self-reporting pipelines, which may add an LLM judge as a final fallback after rule-based parsing or evaluating. Such failures account for less than 1% in every pool in our experiments and have minimal impact on the reported experimental precision.

The open-weight PE control averages negative log-probabilities of sampled tokens over a continuation, rather than computing full-vocabulary entropy at each step. The OpenAI path does not expose this

Table 12: **Hyperparameters used in our experiments.** Top block: decoding configuration for model inference. Bottom block: active-testing hyperparameters of our method.

Model inference		
Parameter	Value	Description
k	10	Surrogate generations per input
`temperature`	0.7	Sampling temperature
`top_p`	0.8	Nucleus sampling threshold
`top_k`	20	Top-k sampling cutoff
`presence_penalty`	1.5	Presence penalty
`repetition_penalty`	1.0	Repetition penalty
`max_new_tokens`	model-max	Maximum supported by each model
Active testing (our method)		
Parameter	Value	Description
H	5	Number of semantic-entropy strata
δ	0.75	Proxy-Neyman allocation offset (Eq. (3))
T	3,000	Monte Carlo trials per labeling budget

Table 13: Benchmarks used in our experiments.

Benchmark	Task type	Knowledge range	Pool size
MMLU-Redux (Gema et al., 2025)	Language	57 academic subjects (cleaned MMLU (Hendrycks et al., 2020))	3,000
MMLU-Pro (Wang et al., 2024b)	Language	STEM-heavy multi-domain reasoning	12,032
MMBench (Liu et al., 2024)	Vision-language	Multimodal perception and reasoning	4,329
GPQA-Diamond (Rein et al., 2023)	Language	Graduate-level science reasoning	198

signal, so its zero placeholders must not be treated as measured PE. API generation is also not made deterministic by the locally recorded inference seed.

Active-testing hyperparameters. Unless otherwise stated, our method uses the hyperparameters summarized in Table 12. We describe the parameter-selection process in Section 5.3.

Allocation implementation details. The nominal $H = 5$ includes the zero-SE stratum and four positive-SE quantile bins; empty bins are omitted from estimation, and repeated quantile boundaries can yield fewer nonempty strata. Given their sizes N_h, the implementation reserves one sample per nonempty stratum. It normalizes the positive allocation scores to divide the remaining budget, takes integer floors, caps counts at N_h, and distributes residual samples in descending remainder order while skipping full strata. A budget smaller than the number of nonempty strata is infeasible. All-zero scores fall back to size-proportional weights. This constrained integer rule is the implemented version of the proportionality in Eq. (3), including for the Oracle-Neyman diagnostic.